

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant(s):	Rabindranath Dutta, Karthikeyan Ramamoorthy		
Assignee:	International Business Machines Corporation		
Title:	System and Process for Enhancing Method Calls of Special Purpose Object-Oriented Programming Languages to Have Security Attributes for Access Control		
Serial No.:	09/817,102	Filing Date:	March 26, 2001
Examiner:	Syed Zia	Group Art Unit:	2131
Docket No.:	AUS920010044US1	Customer No.	65362

---

Austin, Texas  
July 17, 2008

FILED ELECTRONICALLY

**PRE-APPEAL BRIEF REQUEST FOR REVIEW AND STATEMENT OF REASONS**

Sir:

Applicants request review of the Final Office Action in this application. No amendments are being filed with the request. This request is being filed with a Notice of Appeal. The following sets forth a succinct, concise, and focused set of arguments for which the review is being requested.

**CLAIM STATUS**

Claims 1-38 are pending. Claims 1-38 are rejected under 35 U.S.C. §102(e) as being anticipated by U.S. Patent No. 6,044,373 to Gladney et al. (hereinafter “Gladney”).

**REMARKS**

Applicants’ invention relates to generating and using an enforcement construct within a special purpose object-oriented programming language in order to control access to a protected method. More specifically, as set forth in previously amended independent Claims 1, 9, 12, 15, 16, 24, 27, 30, and 31, “an instruction to enforce said object-oriented enforcement construct is embedded in compiled source code and said source code is stored in a computer-readable medium.” Accordingly, within source code statements, the enforcement construct comprises an enforcement keyword (the “instruction”) that indicates an authorization restriction on the invocation of an object-oriented method in conjunction with an identifier of an authorization mechanism, such as an authorization method. In the runtime environment, when a call is initiated to an object-oriented

method, a check is made as to whether the object-oriented method has been protected by an enforcement construct. If so, then the identifier of the associated authorization method is used to invoke the authorization method, which determines whether an entity that is attempting to call or invoke the object-oriented method is authorized to execute the object-oriented method. If so, then the object-oriented method is invoked, and if not, an error response may be returned to the calling entity. The enforcement construct may be applied at the class level such that each method defined within a class becomes a protected method.

The Gladney reference is directed to a system and method for controlling a client's access to a protected element, wherein the protected element is contained in a protected resource which includes a data manager. In Gladney, the protecting resource and the protected resource are arranged in a distributed configuration.

Applicants agree with Examiner that Gladney teaches a method and computer program product for object oriented access control. However, Applicants respectfully disagree with Examiner that Gladney teaches the application of an enforcement construct at the class level such that each method defined within a class becomes a protected method. While the authorization process of Gladney enforces access to objects and protected methods, authorization is enforced at an application level with the data manager making calls to the protecting resource manager. In contrast, the present invention enforces authorization at the object level.

As an example, a first code portion defines class "ClientClassA" that contains method "A" that invokes method "M" in class "ServerClassM" and a second code portion defines a class "ClientClassB" that contains method "B" that likewise invokes method "M" in class "ServerClassM." The first code portion, comprising class "ClientClassA," is compiled into a first application or module while the second code portion, comprising class "ClientClassB," is compiled into a second application or module. The first application or module is used by a user with a first authority role and the second application or module is used by a user with a second authority role. At some point in time, the first user requests some type of action that causes the execution of method "A" within class "ClientClassA." Likewise, at some point in time, the second user requests some type of action that causes the execution of method "B" within class "ClientClassB." Both method "A" and method "B" will attempt to invoke method "M" of class "ServerClassM." When method "M" of class "ServerClassM" is invoked, an exception might be thrown.

More specifically, when method "A" and method "B" attempt to invoke method "M" of class "ServerClassM," it is not possible to predict whether protected method "M" will execute without reference to the state of the runtime environment. The request to invoke method "M" of class

"ServerClassM" might result in the throwing of an exception, depending upon the result of the authorization mechanism, which relies upon predetermined authorization information. In this example, method "M" is protected by enforcing an authorization process defined by method "AuthMethod" of class "ManagerAuthorization." Hence, the invocation of method "M" within method "A" will successfully execute because method "M" was invoked on behalf of the first user, who has been authenticated as being one of a set of authorized users, as determined by the "ManagerAuthorization" mechanism. The invocation of method "M" within method "B" will not cause the execution of method "M" because method "M" was invoked on behalf of the second user, who has been determined by the "ManagerAuthorization" mechanism to not be one of a set of authorized users.

Applicants respectfully submit that this feature is not taught by Gladney, nor any other art of record. Applicants further submit that Gladney fails to provide a teaching of an object-oriented enforcement construct associated with the invocation of a method, as recited in the aforementioned independent claims.

In view of the foregoing, it is respectfully submitted that independent claims 1, 9, 12, 15, 16, 24, 27, 30, and 31 are allowable over the art of record. Furthermore, the pending dependent claims are allowable as being dependent upon allowable base claims.

### **CONCLUSION**

In view of the remarks set forth herein, the application is believed to be in condition for allowance and a notice to that effect is solicited. Nonetheless, should any issues remain that might be subject to resolution through a telephonic interview, the examiner is requested to telephone the undersigned at 512-338-9100.

#### **CERTIFICATE OF TRANSMISSION**

I hereby certify that on July 17, 2008 this correspondence is being transmitted via the U.S. Patent & Trademark Office's electronic filing system.

*/Gary W. Hamilton/*

Respectfully submitted,

*/Gary W. Hamilton/*

Gary W. Hamilton  
Attorney for Applicant(s)  
Reg. No. 31,834